



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Teori Umum

##### 2.1.1. Pengertian Komputer

Dalam bukunya, Sibero (2013:9) mengungkapkan bahwa Komputer berasal dari kata “*Compute*” yang juga berasal dari bahasa latin “*Computare*” yang artinya menghitung”.

Menurut Daulay (2007:17), “Komputer didefinisikan sebagai sekumpulan alat logik yang dapat menerima data, mengolah data, dan menyimpan data dengan menggunakan program yang terdapat pada memori sistem komputer kemudian memberikan hasil pengolahan tersebut dalam bentuk *output*”.

Jogiyanto (1995:2), menyimpulkan bahwa “Komputer adalah alat elektronik yang dapat menerima input data, mengolah data, dan memberikan informasi dengan menggunakan suatu program yang tersimpan di memori komputer (*Stored Program*) serta dapat menyimpan program dan hasil pengolahan yang bekerja secara otomatis”.

Maka dapat disimpulkan bahwa komputer adalah alat elektronik yang mampu menerima data, mengolah data, menyimpan serta menghasilkan *output* dari pengolahan tersebut.

##### 2.1.2. Pengertian Data

Menurut Longkutoy (1998:4), “Data adalah bagian dari fakta yang mengandung arti yang di hubungkan dengan kenyataan, simbol-simbol, gambar-gambar, kata-kata, angka-angka, huruf-huruf atau simbol yang menunjukkan suatu ide, objek, kondisi, atau situasi dan lain-lain”.

Kristanto (2011:7) menyatakan bahwa, “Data menggambarkan suatu kejadian yang sedang terjadi, dimana data tersebut akan diolah dan diterapkan dalam sistem menjadi input yang berguna dalam suatu sistem”.



Menurut Romney dan Steinbart (2016:4), “Data adalah fakta yang dikumpulkan, disimpan, dan diproses oleh sistem informasi”.

Al Fatta (2007:9), “Data merupakan nilai, keadaan, atau sifat yang berdiri sendiri lepas dari konteks apapun”.

Dapat disimpulkan bahwa data adalah gambaran dari suatu kejadian-kejadian nyata yang terjadi yang menunjukkan ide atau objek, kemudian data ini akan diolah menjadi bentuk yang lebih berarti.

### **2.1.3. Pengertian Aplikasi**

Menurut Febrian (2004:35), “Aplikasi adalah program siap pakai atau program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain”.

Menurut Hendrayudi (2009:43), ”Aplikasi adalah kumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus)”.

Budiharto (2013:5) menjelaskan bahwa, “Aplikasi merupakan program yang dapat berjalan di komputer tersendiri (*stand alone computer*), dari mulai program yang simpel sampai dengan program besar dan rumit”.

Jadi, aplikasi adalah program yang di jalankan di dalam komputer dengan bantuan sistem operasi untuk menjalankannya, program ini dibuat dengan fungsi dan kegunaannya masing-masing.



## **2.1.4. Tahapan Analisis Sistem**

### **2.1.4.1. Penyelidikan Awal**

Menurut Kristanto (2011:39) , pada tahap penyelidikan awal, analis belajar dari pemakai mengenai apa yang diharapkan dari sebuah sistem informasi yang baru. Hal-hal yang perlu diperhatikan dalam tahapan ini adalah

1. Mencoba memahami dan memperjelas apa yang diharapkan oleh pemakai (sistem informasi yang bagaimana yang mereka perlukan)
2. Menentukan ruang lingkup dari studi sistem informasi
3. Menentukan kelayakan dari masing-masing alternatif dengan memperkirakan keuntungan / kerugian yang didapat.

### **2.1.4.2. Studi Kelayakan**

Al Fatta (2007:), “Studi kelayakan juga disebut sebagai tahapan analisis kelayakan. Dokumen yang dihasilkan dari tahapan-tahapan sebelumnya dikumpulkan menjadi suatu proposal pendahuluan proyek. Untuk memastikan usulan tersebut bisa diteruskan menjadi proyek yang menguntungkan maka proposal proyek harus dievaluasi kelayakannya dari berbagai segi kelayakan, di antaranya :

1. Kelayakan teknis
2. Kelayakan operasional
3. Kelayakan ekonomi
4. Kelayakan hukum.

## **2.1.5. Metode Pengembangan Sistem**

Sukanto dan Shalahuddin (2013:28) menjelaskan tentang metode pengembangan sistem yaitu *waterfall*. Metode air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut mulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).



a. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasi kebutuhan perangkat lunak agar dapat dipahami, perangkat lunak seperti apa yang dibutuhkan oleh *user*.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengkodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

c. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

d. Pengujian

Pengujian fokus pada perangkat lunak dari segi logik dan fungsional serta memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. Pendukung (*support*) atau pemeliharaan (*maintanance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirim ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada.



## 2.2. Teori Judul

### 2.2.1. Pengertian Sistem

Menurut Al Fatta (2007 : 3) mendefinisikan “Sistem adalah suatu kumpulan dari bagian –bagian yang bekerja sama untuk mencapai tujuan yang sama atau sekumpulan objek-objek yang saling berelasi dan berinteraksi serta hubungan antar objek bisa dilihat satu kesatuan yang dirancang untuk mencapai satu tujuan”.

Sistem terdiri dari unsur-unsur seperti masukan (*input*), pengolahan (*processing*), serta keluaran (*output*) (Fatta, 2007:4).

Menurut Indrajit (2001:2) dalam buku Hutahaeon (2014:1) mengemukakan bahwa “Sistem mengandung arti kumpulan- kumpulan dari komponen- komponen yang dimiliki unsur keterkaitan antara satu dengan lainnya”.

Maka dari itu dapat di simpulkan bahwa sistem adalah suatu kumpulan dari elemen-elemen yang saling terorganisasi, berinterkasi, dan saling bergantung satu sama lain untuk mencapai tujuan tertentu

### 2.2.2. Pengertian Infomasi

Menurut Sibero (2013:10) menjelaskan “Informasi adalah sekumpulan data yang memiliki maksud dan tujuan serta dapat memberikan keterangan akurat yang diperlukan dalam penga mbilan keputusan”.

Romney dan Steinbart (2016:4), “Informasi (*information*) adalah data yang telah dikelola dan diproses untuk memberikan arti dan memperbaiki proses pengambilan keputusan”.

Menurut Davis (1995) dalam buku Al Fatta (2007:9) , “Informasi adalah data yang telah diolah menjadi bentuk yang lebih berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang ”.

Jadi berdasarkan pengertian-pengertian diatas, disimpulkan bahwa informasi adalah sekumpulan data-data yang telah diolah menjadi bentuk yang lebih berguna untuk pengambilan keputusan.



### **2.2.3. Pengertian Penjualan**

Hidayat (2013:8) menjelaskan mengenai pengertian penjualan yaitu suatu transaksi antara penjual dan pembeli dengan kesepakatan kedua belah pihak berupa harga dan barang setelah disetujui bersama.

Menurut Sujarweni (2015:79), "Penjualan adalah suatu sistem kegiatan pokok perusahaan untuk memperjual- belikan barang dan jasa yang perusahaan hasilkan".

Maka dapat disimpulkan menurut penulis, penjualan adalah suatu kegiatan transaksi jual beli barang atau jasa dengan kesepakatan harga yang telah disetujui.

### **2.2.4. Pengertian Penjualan Tunai dan Kredit**

Menurut Sujarweni (2015:79), terdapat dua macam sistem penjualan, yaitu sistem penjualan tunai dan sistem penjualan kredit. Sistem penjualan tunai merupakan sistem yang diberlakukan oleh perusahaan dalam menjual barang dengan cara mewajibkan pembeli untuk melakukan pembayaran harga terlebih dahulu sebelum barang diserahkan pada pembeli.

Sujarweni (2015:89), "Sistem penjualan kredit merupakan sistem penjualan di mana pembayarannya dilakukan setelah barang diterima pembeli, jumlah dan jatuh tempo pembayarannya disepakati oleh kedua pihak".

### **2.2.5. Pengertian Sistem Informasi Penjualan *Interior* Rumah Tunai dan Kredit pada CV. Madian Jaya Abadi**

Sistem Informasi Penjualan *Interior* Rumah Tunai dan Kredit pada CV. Madian Jaya Abadi adalah program berbasis *web* yang dibuat untuk memudahkan admin untuk mengelolah data-data penjualan baik secara tunai maupun kredit dan membantu pelanggan kredit melakukan pembayaran angsuran melalui *web* ini dan membantu pelanggan kredit untuk melihat dan mencetak rincian pembayaran angsuran. Dengan adanya aplikasi ini, diharapkan penyampaian informasi dapat lebih cepat, tepat, akurat dan terhindar dari tindak penipuan atau penyalagunaan oleh pihak tertentu.



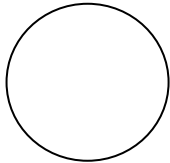
## 2.3. Teori Khusus

### 2.3.1. Data Flow Diagram (DFD)

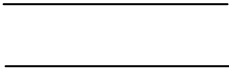

Sukamto dan Shalahuddin (2014:70), “*Data Flow Diagram (DFD)* atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*)”.

Sukamto dan Shalahuddin (2014:71) menjelaskan notasi pada DFD adalah sebagai berikut :

**Tabel 2.1.** Simbol-Simbol *Data Flow Diagram (DFD)*

No.	Notasi	Keterangan
1.		<p>Proses atau fungsi atau prosedur; pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya menjadi fungsi atau prosedur di dalam kode program</p> <p>catatan: nama yang diberikan pada sebuah proses biasanya berupa kata kerja</p>


Lanjutan **Tabel 2.1.** Simbol-Simbol *Data Flow Diagram (DFD)*

No.	Notasi	Keterangan
2.		<p><i>File</i> atau basis data atau penyimpanan (<i>storage</i>); pada pemodelan perangkat lunak yang akan diimplementasikan dengan pemrograman terstruktur, maka pemodelan notasi inilah yang harusnya dibuat menjadi tabel-tabel basis data yang dibutuhkan, tabel-tabel ini juga harus sesuai dengan perancangan tabel-tabel pada basis data (<i>Entity Relationship Diagram (ERD)</i>, <i>Conceptual Data Model(CDM)</i>, <i>Physical Data Model (PDM)</i>)</p> <p>catatan : nama yang diberikan pada sebuah penyimpanan biasanya kata benda</p>
3.		<p>Entitas luar (<i>external entity</i>) atau masukan (<i>input</i>) atau keluaran (<i>output</i>) atau orang yang memakai/berinteraksi dengan perangkat lunak yang dimodelkan atau sistem lain yang terkait dengan aliran data dari sistem yang dimodelkan</p> <p>catatan : nama yang digunakan pada masukan (<i>input</i>) atau keluaran (<i>output</i>) berupa kata benda</p>

(Sumber: Sukamto dan shalahuddin, 2014:71)



Lanjutan **Tabel 2.1.** Simbol-Simbol *Data Flow Diagram (DFD)*

No.	Notasi	Keterangan
4.		Aliran data; merupakan data yang dikirim antar proses, dari penyimpanan ke proses, atau dari proses ke masukan ( <i>input</i> ) atau keluaran ( <i>output</i> )  catatan : nama yang digunakan pada aliran data biasanya berupa kata benda, dapat diawali dengan kata data misalnya “da ta siswa” atau tanpa kata data misalnya “siswa”

(Sumber: Sukanto dan shalahuddin, 2014:71)

Sukanto dan Shalahuddin (2014:72) menjelaskan tentang tahapan-tahapan perancangan dengan menggunakan DFD:

1. Membuat DFD Level 0 atau sering disebut juga *Context Diagram*

DFD Level 0 menggambarkan sistem yang akan dibuat sebagai suatu entitas tunggal yang berinteraksi dengan orang maupun sistem lain. DFD Level 0 digunakan untuk menggambarkan interaksi antara sistem yang akan dikembangkan dengan entitas luar.

2. Membuat DFD Level 1

DFD Level 1 digunakan untuk menggambarkan modul-modul yang ada dalam sistem yang akan dikembangkan. DFD Level 1 merupakan hasil *breakdown* DFD Level 0 yang sebelumnya sudah dibuat.

3. Membuat DFD Level 2

Modul-modul pada DFD Level 1 dapat di-*breakdown* menjadi DFD Level 2. Modul mana saja yang harus di-*breakdown* lebih detail tergantung pada kedetailan modul tersebut. Apabila modul tersebut sudah cukup detail dan rinci maka modul tersebut sudah tidak perlu di-*breakdown* lagi. Untuk sebuah sistem, jumlah DFD Level 2 sama dengan jumlah modul pada DFD Level 1 yang di-*breakdown*.



#### 4. Membuat DFD Level 3 dan seterusnya


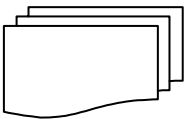
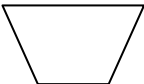
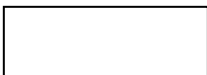
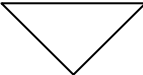
DFD Level 3,4,5, dan seterusnya merupakan *breakdown* dari modul pada DFD Level di-atasnya. *Breakdown* pada level 3,4,5, dan seterusnya aturannya sama persis dengan DFD Level 1 atau 2.

#### 2.3.2. Block Chart

Kristanto (2008:75) mengemukakan, “*Block chart* berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol-simbol tertentu. Pembuatan *block chart* harus memudahkan bagi pemakai dalam memahami alur dari sistem atau transaksi”.

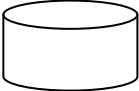

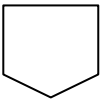
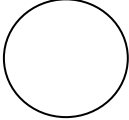
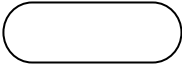
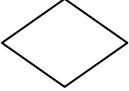
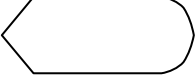

Adapun simbol-simbol *Block Chart* adalah sebagai berikut:

**Tabel 2.2.** Simbol-Simbol *Block Chart*

No.	Simbol	Keterangan
1.		Menandakan dokumen, bisa dalam bentuk surat, formulir, buku/bendel/berkas atau cetakan
2.		Multi dokumen
3.		Proses manual
4.		Proses yang dilakukan oleh computer
5.		Menandakan dokumen yang diarsifkan (arsip manual)

(Sumber: Kristanto, 2008:64)

Lanjutan **Tabel 2.2.** Simbol-Simbol *Block Chart*

No.	Simbol	Keterangan
6.		Data penyimpanan ( <i>data storage</i> )
7.		Proses apa saja yang tidak terdefinisi termasuk aktifitas fisik
8.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang lain
9.		Terminasi yang mewakili simbol tertentu untuk digunakan pada aliran lain pada halaman yang sama
10.		Terminasi yang menandakan awal dan akhir dari suatu aliran
11.		Pengambilan keputusan ( <i>decision</i> )
12.		Layar peraga ( <i>monitor</i> )
13.		Pemasukan data secara manual





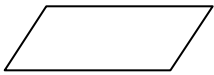
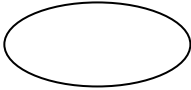
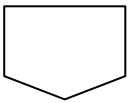
(Sumber: Kristanto, 2008:64)



### 2.3.3. Flowchart


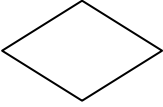




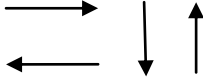


Supardi (2013:51), “Bagan alir (*flowchart*) merupakan bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi”. Adapun simbol-simbol *Flowchart* adalah sebagai berikut:

**Tabel 2.3.** Simbol-Simbol *Flowchart*

No.	Simbol	Arti
1.		Simbol titik terminal ( <i>terminal point symbol</i> ) digunakan untuk menunjukkan awal dan akhir dari suatu proses.
2.		Simbol proses digunakan untuk mewakili suatu proses.
3.		Simbol proses terdefinisi ( <i>predefined process symbol</i> ) digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan di tempat lain.
4.		Simbol persiapan ( <i>preparation symbol</i> ) yang digunakan untuk memberi nilai awal suatu besaran.
5.		Simbol input atau output digunakan untuk mewakili data input atau output.
6.		Simbol penghubung ( <i>connector symbol</i> ) digunakan untuk menunjukkan sambungan dari alir yang terputus di halamanyang masih sama.
7.		Simbol penghubung ( <i>connector symbol</i> ) digunakan untuk menunjukkan sambungan dari alir yang terputus ke halaman lainnya.

(Sumber: Supardi, 2013:53)

Lanjutan Tabel 2.3. Simbol-Simbol *Flowchart*

No.	Simbol	Arti
8.		Menyatakan pencetakan (dokumen) pada kertas.
9.		Menyatakan <i>desicion</i> (keputusan) yang digunakan untuk penyeleksian kondisi di dalam program.
10.		Menyatakan media penyimpanan drum magnetik.
11.		Menyatakan <i>input/output</i> menggunakan disket.
12.		Menyatakan operasi yang dilakukan secara manual.
13.		Menyatakan <i>input/output</i> dari kartu plong.
14.		Menyatakan arah aliran pekerjaan (proses).
15.		<i>Multidocument</i> (banyak dokumen).
16.		<i>Delay</i> (penundaan atau kelambatan).

(Sumber: Supardi, 2013:53)

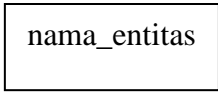
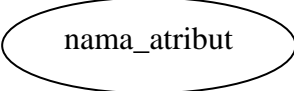
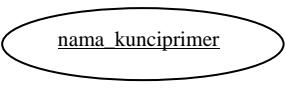
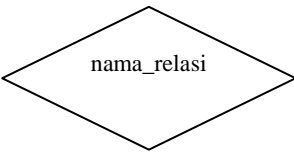


### 2.3.4. Entity Relationship Diagram (ERD)

Sukamto dan Shalahuddin (2014:50), “Entity Relationship Diagram (ERD) adalah pemodelan awal basis data yang dikembangkan berdasarkan teori himpunan dalam bidang matematika untuk pemodelan basis data relational”.

Sukamto dan Shalahuddin (2014:50), menjelaskan simbol-simbol yang digunakan dalam ERD dengan notasi Chen, yaitu:

**Tabel 2.4.** Simbol-Simbol *Entity Relationship Diagram (ERD)*

No	Simbol	Keterangan
1.	Entitas/ <i>entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
3.	Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)
4.	Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja

(Sumber: Sukamto dan Shalahuddin, 2014:50-51)

Lanjutan **Tabel 2.4.** Simbol-Simbol *Entity Relationship Diagram (ERD)*

No	Simbol	Keterangan
5.	Asosiasi / <i>association</i>  _____	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian  Kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas. Misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B

(Sumber: Sukamto dan Shalahuddin, 2014:50-51)

### 2.3.5. Kamus Data (*Data Dictionary*)

Sukamto dan Shalahuddin (2014:73) mengemukakan, “Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar penulisan)”. Kamus data biasanya berisi:

1. Nama – nama dari data
2. Digunakan pada – merupakan proses-proses yang terkait data
3. Deskripsi – merupakan deskripsi data
4. Informasi tambahan – seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut:

**Tabel 2.5.** Simbol-Simbol Kamus Data

No.	Simbol	Keterangan
1.	=	disusun atau terdiri dari
2.	+	Dan
3.	[   ]	baik ...atau...

Lanjutan **Tabel 2.5.** Simbol-Simbol Kamus Data

4.	{ } <sup>n</sup>	n kali diulang / bernilai banyak
5.	( )	data opsional
6.	*...*	batas komentar

(Sumber: Sukamto dan Shalahuddin, 2014:7 4)

### 2.3.6. Daftar Kejadian (*Event List*)

Kristanto (2008:70) menjelaskan tentang daftar kejadian (*event list*) sebagai berikut:

Daftar kejadian digambarkan dalam bentuk kalimat sederhana dan berfungsi untuk memodelkan kejadian yang terjadi dalam lingkungan sehari-hari dan membutuhkan tanggapan atau respon dari sistem. Suatu kejadian mewakili satu aliran data atau proses dalam diagram konteks serta deskripsi penyimpanan yang digunakan untuk memodelkan data harus diperhatikan dalam kaitannya dengan daftar kejadian.

Adapun cara-cara mendeskripsikan daftar kejadian adalah sebagai berikut:

1. Pelaku adalah entiti luar, jadi bukan sistem.
2. Menguji setiap entiti luar dan mencoba mengevaluasi setiap entiti luar yang terjadi pada sistem.
3. Hati-hati dengan kejadian yang spesifik, yang tak sengaja menyatu dalam paket yang sama.
4. Harus diingat bahwa kejadian yang dimodelkan bukan hanya interaksi normal antara sistem dengan entiti luar, karena itu harus dievaluasi kebutuhan sistem untuk menanggapi kejadian yang gagal.
5. Setiap aliran keluaran sebaiknya merupakan respondari kejadian.
6. Setiap kejadian yang tidak berorientasi pada waktu dalam daftar kejadian sebaiknya mempunyai masukan sehingga sistem dapat mendeteksi kejadian yang berlangsung.
7. Setiap kejadian sebaiknya menghasilkan keluaran langsung sebagai respon atau disimpan dalam berkas untuk bahan masukan.





## 2.4. Teori Program

### 2.4.1. PHP (*PHP Hypertext Preprocessor*)

#### 2.4.1.1. Pengertian *PHP*

Sibero (2013:49) mengemukakan, “*PHP* adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan”.

Nugroho (2013:153) mengemukakan, “*PHP* itu bahasa pemrograman berbasis *web*. Jadi, *PHP* itu adalah bahasa program yang digunakan untuk membuat aplikasi berbasis *web* (*website*, *blog*, atau aplikasi *web*).

Maka dapat disimpulkan bahwa *PHP* merupakan bahasa pemrograman *web* yang berupa *script-script* penerjemah menjadi bahasa yang dapat dimengerti oleh komputer.

#### 2.4.1.2. Mengenal Script *PHP*

Madcoms (2009:133) mengemukakan, “Cara penulisan script atau code program *PHP* adalah diawali dengan pembuka script `<? Atau <?php`, dan diakhiri dengan script penutup`?>`”. Contoh penulisan script *PHP* adalah sebagai berikut:

**Tabel 2.6.** Script Dasar *PHP*

```
<? ... ?>
<? .....?>
<script language="PHP"> ... </script>
```

(Sumber : Madcoms, 2009:133)

Kode program *PHP* dapat disisipkan di bagian manapun dalam *HTML* atau *XHTML*. Halaman yang ada script *PHP* di dalamnya, harus disimpan dengan ekstensi **.php**.



### 2.4.2. Basis Data (*Database*)

Menurut Andi (2005:2), “*Database* adalah sekumpulan data yang terdiri dari satu atau lebih tabel yang saling berhubungan. Anda atau user mempunyai wewenang untuk mengakses data tersebut, baik untuk menambah, mengubah, atau menghapus data yang ada dalam tabel tersebut”.

### 2.4.3. *MySQL*

#### 2.4.3.1. Pengertian *MySQL*

Sibero (2013:97) mengemukakan, “*MySQL* atau dibaca “*My Sekuel*” dengan adalah suatu RDBMS (*Relational Database Management System*) yaitu aplikasi sistem yang menjalankan fungsi pengolahan data”.

Sadeli (2014:10) menjelaskan , “*MySQL* adalah database yang menghubungkan script php menggunakan perintah query dan escapes character yang sama dengan php”.

Kadir (2008:2) menerangkan bahwa, “*MySQL* (baca: mai-se-kyu-el) merupakan *software* yang tergolong DBMS (*Database Management System*) yang bersifat *Open Source*. *Open Source* menyatakan bahwa *software* ini dilengkapi dengan *source code* (kode yang dipakai untuk membuat *MySQL*), selain tentu saja bentuk *executable*-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan bisa diperoleh dengan cara *men-download* (mengunduh) di *internet* secara gratis”.

#### 2.4.3.2. Fungsi-Fungsi *MySQL*

Kadir (2008:360) menjelaskan tentang fungsi yang berawalan **mysql\_** yang digunakan untuk mengakses *database server MySQL* yang dibahas di bawah ini:

**Tabel 2.7.** Fungsi-Fungsi *MySQL*

No	Nama Fungsi	Kegunaan	Bentuk Pemanggilan
1.	<i>MySQL_connect()</i>	Membuat hubungan ke <i>database MySQL</i> yang terdapat pada suatu <i>host</i>	<code>mysql_connect</code> ( <i>host</i> , <i>nama_pemakai</i> , <i>password</i> )

Lanjutan **Tabel 2.7.** Fungsi-Fungsi *MySQL*

No	Nama Fungsi	Kegunaan	Bentuk Pemanggilan
2.	<i>MySQL_close()</i>	Menutup hubungan ke <i>database MySQL</i>	<code>mysql_close</code> ( <code>pengenal_hubungan</code> )
3.	<i>MySQL_select_db()</i>	Memilih database	<code>mysql_select_db</code> ( <code>database</code> , <code>pengenal_hubungan</code> )
4.	<i>MySQL_query()</i>	Mengeksekusi permintaan terhadap sebuah tabel atau sejumlah tabel	<code>mysql_query</code> ( <code>permintaan</code> , <code>pengenal_hubungan</code> )
5.	<i>MySQL_db_query()</i>	Menjalankan suatu permintaan terhadap suatu <i>database</i>	<code>mysql_db_query</code> ( <i>database</i> , <code>permintaan</code> , <code>pengenal_hubungan</code> )
6.	<i>MySQL_num_rows()</i>	Memperoleh jumlah baris dari suatu hasil permintaan ( <i>query</i> )	<code>mysql_num_rows</code> ( <code>pengenal_hasil</code> )
7.	<i>MySQL_affected_rows()</i>	Memperoleh jumlah baris yang dikenai operasi INSERT, DELETE, dan UPDATE	<code>mysql_affected_rows</code> ( <code>[pengenal_hubungan]</code> )
8.	<i>MySQL_num_fields()</i>	Memperoleh jumlah kolom pada suatu hasil permintaan	<code>mysql_num_fields</code> ( <code>pengenal_hasil</code> )
9.	<i>MySQL_fetch_row()</i>	Menghasilkan suatu <i>array</i> yang berisi seluruh kolom dari sebuah baris pada suatu himpunan hasil	<code>mysql_fetch_row</code> ( <code>pengenal_hasil</code> )

Lanjutan **Tabel 2.7.** Fungsi-Fungsi *MySQL*

No	Nama Fungsi	Kegunaan	Bentuk Pemanggilan
10.	<i>MySQL_fetch_array()</i>	Kegunaannya serupa dengan <i>mysql_fetch_row()</i> . Hanya saja, setiap kolom akan disimpan dua kali pada <i>array</i> hasil	<i>mysql_fetch_array</i> ( <i>pengenal_hasil</i> )
11.	<i>MySQL_fetch_field()</i>	Memperoleh informasi suatu kolom	<i>mysql_fetch_field</i> ( <i>pengenal_hasil</i> [ , <i>nomor_kolom</i> ])
12.	<i>MySQL_data_seek()</i>	Memindah <i>pointer</i> pada suatu himpunan hasil supaya menunjuk ke baris tertentu	<i>mysql_data_seek</i> ( <i>pengenal_hasil</i> , <i>nomor_baris</i> )
13.	<i>MySQL_field_seek()</i>	Memindah <i>pointer</i> pada suatu himpunan hasil supaya menunjuk ke kolom tertentu	<i>mysql_data_seek</i> ( <i>pengenal_hasil</i> , <i>nomor_kolom</i> )
14.	<i>MySQL_create_db()</i>	Menciptakan <i>database MySQL</i>	<i>mysql_create_db</i> ( <i>database</i> [ , <i>pengenal_hubungan</i> ])
15.	<i>MySQL_drop_db()</i>	Menghapus <i>database MySQL</i>	<i>mysql_drop_db</i> ( <i>database</i> [ , <i>pengenal_hubungan</i> ])
16.	<i>MySQL_list_dbs()</i>	Menghasilkan daftar <i>database MySQL</i>	<i>mysql_list_dbs</i> ( [ <i>pengenal_hubungan</i> ] )

Lanjutan **Tabel 2.7.** Fungsi-Fungsi *MySQL*

No	Nama Fungsi	Kegunaan	Bentuk Pemanggilan
17.	<i>MySQL_list_tables</i> ( )	Memperoleh daftar nama tabel dalam suatu <i>database</i>	<i>mysql_list_tables</i> ( <i>database</i> [ , <i>pengenal_hubungan</i> ])
18.	<i>MySQL_list_fields</i> ( )	Memperoleh daftar nama kolom dalam suatu <i>database</i>	<i>mysql_list_fields</i> ( <i>database</i> [ <i>pengenal_hubungan</i> ])

(Sumber: Kadir, 2008:360-379)

#### 2.4.4. Adobe Dreamweaver CS5

##### 2.4.4.1. Pengertian Adobe Dreamweaver CS5

Wahana (2009:2), “*dreamweaver* adalah salah satu aplikasi yang dapat dimanfaatkan untuk merancang desainweb secara visual atau aplikasi editor web”.

Pada *dreamweaver CS5*, terdapat beberapa kemampuan bukan hanya sebagai *software* untuk desain *Web* saja tetapi juga untuk menyunting kode serta pembuatan aplikasi *Web* dengan menggunakan berbagai bahasa pemrograman *Web*.



**Gambar 2.1.** Logo *Dreamweaver CS5*



#### 2.4.4.2. Ruang Kerja Dasar *Adobe Dreamweaver CS5*

Ruang kerja dasar *Adobe Dreamweaver CS5* adalah sebagai berikut:

##### 1. *Application Bar*

*Application bar* berada di bagian paling atas jendela aplikasi *Dreamweaver CS5*. Baris ini berisi tombol *Workspace (Workspace Switcher)*, *CS Live*, menu, dan aplikasi lainnya.



**Gambar 2.2.** Tampilan *Application Bar*

##### 2. *Toolbar Document*

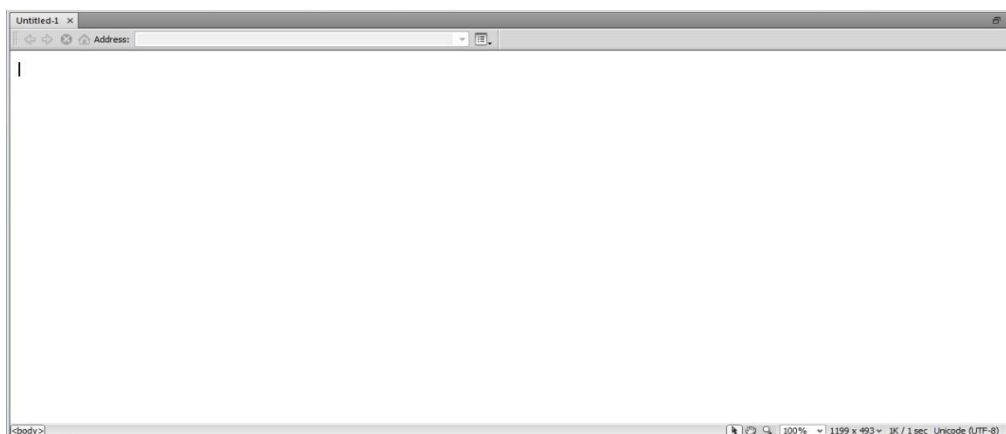
*Toolbar document* berisi tombol-tombol yang berfungsi untuk mengubah tampilan jendela dokumen, sebagai contoh tampilan jendela *Design* atau tampilan *Code*. Juga dapat digunakan untuk operasi-operasi umum, misalnya untuk melihat hasil sementara halaman *web* pada jendela browser.



**Gambar 2.3.** Tampilan *Toolbar Document*

##### 3. Jendela Dokumen

Jendela dokumen berfungsi sebagai lembar kerja untuk membuat dan mengedit desain halaman *web*.

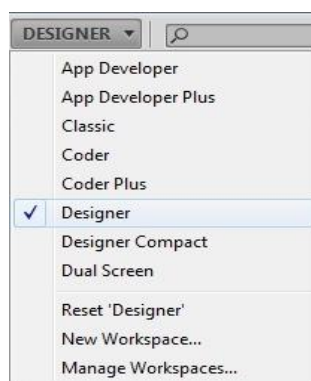


**Gambar 2.4.** Tampilan Jendela Dokumen



#### 4. *Workspace Switcher*

*Workspace switcher* digunakan untuk mengubah tampilan ruang kerja (*workspace*) *Dreamweaver CS5*.



**Gambar 2.5.** Tampilan *Workspace Switcher*

#### 5. *Panel Groups*

*Panel groups* adalah kumpulan panel yang saling berkaitan, panel-panel ini dikelompokkan pada judul-judul tertentu berdasarkan fungsinya. Panel ini digunakan untuk memonitor dan memodifikasi pekerjaan. Secara default, *panel group* berisi panel *Adobe BrowserLab*, *Adobe Business Catalyst*, *Insert*, *CSS Styles*, *Asset*, *AP Element*, dan *Files*.

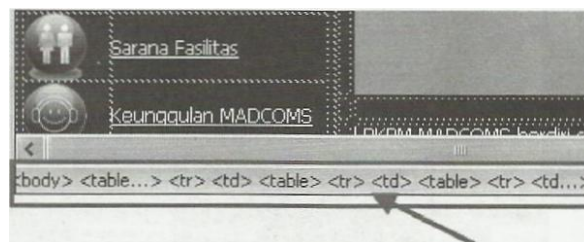


**Gambar 2.6.** Tampilan *Panel Groups*



## 6. *Tag Selector*

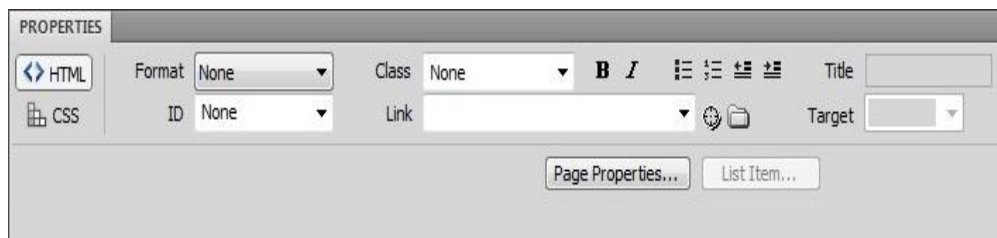
*Tag selector* diletakkan di bagian bawah jendela dokumen. Bagian ini menampilkan hirarki pekerjaan yang sedang terpilih pada jendela dokumen, dapat juga digunakan untuk memilih objek pada jendela desain berdasarkan jenis atau kategori objek tersebut. *Tag selector* juga menampilkan informasi format dari bagian yang sedang aktif pada lembar kerja *Design*.



**Gambar 2.7.** Tampilan *Tag Selector*

## 7. *Property Inspector*

*Property inspector* digunakan untuk melihat dan mengubah berbagai properti objek atau teks yang ada dalam jendela *design*.



**Gambar 2.8.** Tampilan *Property Inspector*

## 8. *Toolbar Standard*

*Toolbar standard*, baris ini berisi tombol-tombol yang mewakili perintah pada menu **File** dan **Edit**, diantaranya perintah **New**, **Open**, **Save**, **Save All**, **Cut**, **Copy**, **Paste**, **Undo**, dan **Redo**. Pilih menu **View** ► **Toolbar** ► **Standard** untuk menampilkannya.



**Gambar 2.9.** Tampilan *Toolbar Standard*





### 9. *Toolbar Style Rendering*

*Toolbar style rendering* secara default disembunyikan. Toolbar ini berisi tombol-tombol untuk menampilkan desain dalam media yang berbeda. Selain itu juga digunakan untuk mengaktifkan dan menonaktifkan style CSS. Untuk menampilkannya, pilih menu **View ► Toolbar ► Style Rendering**.



**Gambar 2.10.** Tampilan *Toolbar Style Rendering*

### 10. *Toolbar Coding*

*Toolbar coding* berisi tombol-tombol yang digunakan untuk melakukan operasi kode-kode standar. Toolbar ini hanya tampil pada jendela *Code*.



**Gambar 2.11.** Tampilan *Toolbar Coding*